Homework 1

b03901027 徐彥旻

Handwriting

1. CIA

Confidentiality

保密性，避免未經授權的人取得資訊。以 google doc 為例，要避免未加入分享的帳號檢視文件。

Integrity

完整性，避免資料有未授權的更改。以 messenger 為例，要確保傳出的訊息跟對方收到的訊息是一致的。

Availability

可用性，（想要使用某服務的）使用者可以順利使用該服務。舉例來說，購物網站限時特價時，會有大量的使用者，同時也可能會有 DDoS 攻擊，若系統能讓使用者順利買到東西，則可用性就有達成（對照的例子是，使用者無法連到購物網站，或是等待的時間過長）。

2. Hash Function

One-wayness

對於任一給定的輸出，難以找到對應的輸入。One-way Hash Chains 即為這個特性的應用，可以用來實作 one-time password，驗證發送方的身分。

Weak collision resistance

對於任一給定的輸入，難以找到相異的輸入，滿足兩者的輸出相同。近來火熱的區塊鍊，即是用此一性質來保證交易紀錄不被竄改。

Strong collision resestance

難以找到相異的輸入，滿足其輸出相同。數位簽章的過程隱含此一性質，簽署者無法預先生成兩份內容，讓經過函數後的結果會相同。因此簽署者無法否認簽章的內容。

3. Symmetric Cryptography with KDC

   a. 如果沒有 N_A，或者 N_A 是常數，攻擊者可以假裝是 KDC，當 A 發出要跟 B 溝通的要求時，將要求攔截，並且回傳之前竊聽到的 $E_{K_{SA}}(K_s || ID_B || E_{K_{SB}} (K_S || ID_A))$，讓 A、B 重複使用同樣的 session key。若 N_B 是常數，攻擊者還可以用之前竊聽到的 $E_{Ks}(f(N_B))$ 傳給 B，使得 B 以為攻擊者是 A。

   b. $E_{K_{SB}}(K_S || ID_A)$ 裡頭並沒有跟時間相關的資訊，只要 K_{SB} 沒有更新，已經畢業的學生 A 仍然可以用之前 KDC 給的 $E_{K_{SB}}(K_S || ID_A)$ 跟 B 建立通訊。又如果其他的攻擊者解出 A 與 B 某一次通訊的 Ks，攻擊者可以用之前竊聽到的 $E_{K_{SB}}(K_S || ID_A)$，偽裝成 A，跟 B 建立通訊。

   c. 提高更新 K_{Si} 的頻率，並在 $E_{K_{SB}}(K_S || ID_A)$ 加入跟時間相關的資訊。舉例來說，將 $E_{K_{SB}}(K_S || ID_A)$ 改成 $E_{K_{SB}}(T || K_S || ID_A)$，其中 T 是跟時間有關的資訊，B 在解開的時候會檢查，如果 T 所代表的時間點是太久以前的，則不與發送方建立通訊。

Capture The Flag

4. Classical Cipher

Flag: BALSN{C14$5ic41_c!ph3r_1$_r34lly_cl455ic41}

第一回合：計算明文的第一個字元減去密文的差距，加在第二個密文上即可。

第二回合：對二十六種位移量的結果，檢查每個字是否出現在英文字典當中，選擇出現最多的送出。

第三回合：觀察明文與密文之間的差距，看其來像是等差級數，還原此等差級數，加到對應的第二個密文上除了空格以外的位置，還原回明文。

第四回合：未解出，但是應可以還原第一對明文密文跟第二個密文重疊的部分，若遇到空格則需查閱字典填入，此方法不適用於第一組明文密文比第二組密文短的情形。

第五回合：紀錄第一對明文密文換位置的情形，對第二個密文做逆向的換位。

第六回合：觀察明文與密文的換位情形，發現是 columnar cipher，確認的 column 數量後，對第二個密文做逆向的換位操作。

第七回合：搜尋提示字串，發現是 base64 編碼，解碼回來就可以了。

5. Google can beat this

Flag: BALSN{D0NT_7RU57_SHA1_N0W}

從網路上得到兩份有相同雜湊值的檔案，截到兩份檔案不再相異的地方為止，得到兩個有一樣雜湊值的兩個足夠短的資料，在後面嘗試接上隨機的資料，直到其雜湊值最後二十四的位元與指定的相同，就做出所需的兩個相異輸入，但其雜湊函數的輸出相同了。

6. Many-time pad

Flag: BALSN{using a key one time is not enough, have you tried using it twice?}

對於每一串密文，與其他九個密文進行互斥或的運算。對於這些結果的每一個位置，若結果皆分布在 upper case, lower case, null 的 ascii 範圍內，則判定那一串密文的那個位置是空白字元。知道了密文跟明文（空白字元）之後，即可解出那個位置的金鑰，然後解開其他九個字串的對應位置的密文。

這樣即可解出大部分的密文，接著再用人工猜測克漏字的方法填補未解出的部分，每次填上一個或數個字元之後，觀察用這些字元去解碼其他字串的結果是否合理，若合理則當作是正確的猜測。如此解出 Flag

7. Backdoor of Diffie Hellman

Flag: BALSN{black magic number}

從題目的提示當中可知，$g_{backdoor}^{691829} \equiv 1 \ mod \ p$，因此若使用 $g_{backdoor}$ 當成生成元，可能的金鑰數量（等於 691829）就會遠小於 p。在可能的金鑰數量這麼少的狀況下，只要使用暴力破解就可以了。

8. Man In The Middle

Flag: BALSN{Wow_you_are_really_in_the_middle}

觀察題目所附的程式碼，發現可能的生成元只有二十個，但由於又分為三個階段，因此最後加密的金鑰為 $g_1^{b_1 a_1} \oplus g_2^{b_2 a_2} \oplus g_3^{b_3 a_3}$，這代表在這三個階段都要猜對生成元才能正確解開密文。故經過 [1, 1, 1] 到 [20, 20, 20] 共八千次的序列窮舉之後，必能取得 Flag.

Acknowledgements

王建元同學

陳佳佑同學

江緯璿同學